<u>CLAIMS</u>

What is claimed is:

1.   A processor-implemented method for compiling high-level blocks of an electronic hardware design in a high-level modeling system (HLMS) into hardware description language (HDL) components, comprising:

establishing clock requirements for the electronic hardware design; and

generating in one pass through the high-level blocks, HDL components that are consistent with the clock requirements.

2. The method of claim 1, further comprising:

establishing explicit connections from implicit connections between the high-level blocks, and wherein the HDL components generated in the one pass are consistent with the explicit connections.

3.   The method of claim 1, further comprising:

traversing depth-first a hierarchy of high-level blocks;

generating, while traversing the hierarchy, associated compiler blocks for the high-level blocks, wherein each compiler block includes data that describes hardware ports associated with a high-level block and a name of an HDL component to be used to implement the compiler block; and

generating, while traversing the hierarchy, an HDL component from a compiler block when an associated high-level block in the hierarchy has no sub-blocks.

4.   The method of claim 3, wherein each compiler block includes a description of one or more hardware ports and a name of an HDL entity to be used to implement the compiler block.

5.   The method of claim 3, wherein the step of propagating includes saving the selected implementation information in association with a parent compiler block of an implemented compiler block.

6.   The method of claim 3, further comprising freeing memory associated with a compiler block after generating the HDL component from the compiler block and before generating an HDL component from another compiler block.

7.   The method of claim 6, wherein:
       the step of generating compiler blocks includes associating a type attribute with each compiler block, wherein the type indicates a type of method by which an HDL component is to be generated from the compiler block; and
       the step of generating the HDL component includes generating the HDL component with the type of method specified by the type attribute associated with the compiler block.

8.   The method of claim 7, further comprising:
       wherein the type attribute includes a core-type attribute; and
       the step of generating the HDL component further includes generating the HDL component from a logic core and user-specified parameters associated with a high-level block.

9.   The method of claim 8, wherein a core-type compiler block includes a name of an HDL entity and instance name to be used for the HDL component, directions of ports, and the user-specified parameters.

10.   The method of claim 7, further comprising:
       wherein the type attribute includes a scripting-type attribute; and

the step of generating the HDL component further
includes applying a predefined script to a predefined HDL
template for a scripting-type compiler block.

11. The method of claim 10, wherein a scripting-type
compiler block includes a name of an HDL entity and instance
name to be used for the HDL component, directions of ports, a
name of an HDL template, and a name of a script.

12. The method of claim 7, further comprising:
    wherein the type attribute includes a netlist-type
attribute; and
    the step of generating the HDL component further
includes generating, for a netlist-type compiler block, an
HDL entity that declares each sub-block as an instance and
generating HDL code that connects the instance.

13. An apparatus for compiling high-level blocks of an
electronic hardware design in a high-level modeling system
(HLMS) into hardware description language (HDL) components,
comprising:
    means for establishing explicit connections from
implicit connections between the high-level blocks; and
    means for generating in one pass through the high-level
blocks, HDL components that are consistent with the global
HDL names and explicit connections.

14. An article of manufacture, comprising:
    an electronically readable medium configured with
instructions for causing a processor to perform the steps
including,
        determining clock requirements in an electronic
    hardware design;
        establishing explicit connections from implicit
    connections between the high-level blocks; and
        generating in one pass through the high-level

blocks, HDL components that are consistent with the
clock requirements and explicit connections.

15. The article of manufacture of claim 14, wherein the
electronically readable medium is configured with further
instructions for causing a processor to perform the steps
comprising:

traversing depth-first a hierarchy of high-level blocks;

generating, while traversing the hierarchy, associated
compiler blocks for the high-level blocks, wherein each
compiler block includes data that describes hardware ports
associated with a high-level block and a name of an HDL
component to be used to implement the compiler block;

generating, while traversing the hierarchy, an HDL
component from a compiler block when the associated high-
level block in the hierarchy has no high-level sub-blocks;
and

propagating selected implementation information f each
HDL component for use in generating related HDL components.

16. The article of manufacture method of claim 15, wherein
each compiler block includes a description of one or more
hardware ports and a name of an HDL entity to be used to
implement the compiler block.

17. The article of manufacture of claim 15, wherein the
electronically readable medium is configured with further
instructions for causing a processor, in propagating, to
perform the step comprising saving the selected
implementation information in association with a parent
compiler block of an implemented compiler block.

18. The article of manufacture of claim 15, wherein the
electronically readable medium is configured with further
instructions for causing a processor to perform the step
comprising freeing memory associated with a compiler block

after generating the HDL component from the compiler block
and before generating an HDL component from another compiler
block.

19.   The article of manufacture of claim 18, wherein the
electronically readable medium is configured with further
instructions for causing a processor to perform the steps
comprising:

in generating compiler blocks, associating a type
attribute with each compiler block, wherein the type
indicates a type of method by which an HDL component is to be
generated from the compiler block; and

in generating HDL components, generating each HDL
component with the type of method specified by the type
attribute associated with the compiler block.

20.   The article of manufacture of claim 19, wherein the type
attribute includes a core-type attribute, and the
electronically readable medium is configured with further
instructions for causing a processor, in generating HDL
components, to perform the step comprising generating an HDL
component from a logic core and user-specified parameters
associated with the high-level block.

21.   The article of manufacture of claim 20, wherein a core-
type compiler block includes a name of an HDL entity and
instance name to be used for the HDL component, directions of
ports, and the user-specified parameters.

22.   The article of manufacture of claim 19, wherein the type
attribute includes a scripting-type attribute, and the
electronically readable medium is configured with further
instructions for causing a processor, in generating HDL
components, to perform the step comprising applying a
predefined script to a predefined HDL template for a
scripting-type compiler block.

23. The article of manufacture of claim 22, wherein a scripting-type compiler block includes a name of an HDL entity and instance name to be used for the HDL component, directions of ports, names of HDL templates, and names of scripts.

24. The article of manufacture of claim 19, wherein the type attribute includes a netlist-type attribute, and the electronically readable medium is configured with further instructions for causing a processor, in generating HDL components, to perform the step comprising generating, for a netlist-type compiler block, an HDL entity that declares each sub-block as an instance and generating HDL code that connects the instances.